Optimal Control on Non-Compact Lie Groups: 
A Projection Operator approach

Alessandro Saccon
Institute for Systems and Robotics, Instituto Superior Técnico, Lisboa

Joint work with Prof. John Hauser and Prof. A. Pedro Aguiar

Padova, 24 May 2010
Introduction

❖ Why do Trajectory Optimization?
❖ Minimization of Trajectory Functionals
❖ Unconstrained (?) Optimal Control
❖ Projection Operator Approach
❖ Projection Operator Properties
❖ Trajectory Manifold
❖ Equivalent Optimization Problems
❖ Projection operator Newton method
❖ Derivatives
❖ Computation of $D^2 P$

Mathematical Preliminaries

Control systems on Lie groups

The Projection Operator approach on Lie groups

Left-trivialized linearization around a trajectory

Projection Operator

Quadratic approximation of the cost function
Why do Trajectory Optimization?

Well known:

- **Optimal control** may be used to provide stabilization, tracking, etc., for **nonlinear** systems

- **Model predictive/receding horizon** strategies have been used successful for a number of **nonlinear** systems with **constraints**
Why do Trajectory Optimization?

Well known:
- **Optimal control** may be used to provide stabilization, tracking, etc., for nonlinear systems
- **Model predictive/receding horizon** strategies have been used successful for a number of nonlinear systems with constraints

Also:
- **Trajectory exploration**: What cool stuff can this system do?
  - capabilities
  - limitations
- **Trajectory modeling**: Can the trajectories of this (complex) system be modeled by those of a simpler system? [e.g., reduced order, flat, ...]
- **Objective function design**: needed to exploit system capabilities
- **Systems analysis**: investigate system structure, e.g., controllability
Consider the problem of minimizing a functional

\[ h(x(\cdot), u(\cdot)) := \int_0^T l(x(\tau), u(\tau), \tau) \, d\tau + m(x(T)) \]

over the set \( \mathcal{T} \) of bounded trajectories of the nonlinear system

\[ \dot{x}(t) = f(x(t), u(t)) \]

with \( x(0) = x_0 \) (... without additional constraints).

We write this **constrained** problem as

\[ \min_{\xi \in \mathcal{T}} h(\xi) \]

where

\( \xi = (\alpha(\cdot), \mu(\cdot)) \) is a bounded curve with \( \alpha(\cdot) \) continuous and \( \alpha(0) = x_0 \).
Minimization of Trajectory Functionals

Consider the problem of minimizing a functional

$$ h(x(\cdot), u(\cdot)) := \int_0^T l(x(\tau), u(\tau), \tau) \, d\tau + m(x(T)) $$

over the set $\mathcal{T}$ of bounded trajectories of the nonlinear system

$$ \dot{x}(t) = f(x(t), u(t)) $$

with $x(0) = x_0$ (... without additional constraints).

We write this constrained problem as

$$ \min_{\xi \in \mathcal{T}} h(\xi) $$

where

$$ \xi = (\alpha(\cdot), \mu(\cdot)) $$

is a bounded curve with $\alpha(\cdot)$ continuous and $\alpha(0) = x_0$.

**How can we approach this problem?**
Introduction

Why do Trajectory Optimization?

Minimization of Trajectory Functionals

Unconstrained (?) Optimal Control

Projection Operator Approach

Projection Operator

Projection Operator Properties

Trajectory Manifold

Equivalent Optimization Problems

Projection operator Newton method

Derivatives

Computation of $D^2 P$

Mathematical Preliminaries

Control systems on Lie groups

The Projection Operator approach on Lie groups

Left-trivialized linearization around a trajectory

Projection Operator

Unconstrained (?) Optimal Control

In the usual case, the choice of a control trajectory $u(\cdot)$ determines the state trajectory $x(\cdot)$ (recall that $x_0$ has been specified). With such a trajectory parametrization, one obtains so-called unconstrained optimal control problem

$$\min_{u(\cdot)} h(x(\cdot; x_0, u(\cdot)), u(\cdot))$$

Why not just search over control trajectories $u(\cdot)$? If the system described by $f$ is sufficiently stable, then such a shooting method may be effective.

Unfortunately, the modulus of continuity of the map $u(\cdot) \mapsto (x(\cdot), u(\cdot))$ is often so large that such shooting is computationally useless:

small changes in $u(\cdot)$ may give LARGE changes in $x(\cdot)$
**Projection Operator Approach**

**Key Idea:** a trajectory tracking controller may be used to minimize the effects of system instabilities, providing a numerically effective, redundant trajectory parametrization.

Let \( \xi(t) = (\alpha(t), \mu(t)), t \geq 0 \), be a bounded curve and let \( \eta(t) = (x(t), u(t)), t \geq 0 \), be the trajectory of \( f \) determined by the nonlinear feedback system

\[
\begin{align*}
\dot{x} & = f(x, u), \\
\dot{u} & = \mu(t) + K(t)(\alpha(t) - x).
\end{align*}
\]

The map

\[
\mathcal{P} : \xi = (\alpha(\cdot), \mu(\cdot)) \mapsto \eta = (x(\cdot), u(\cdot))
\]

is a continuous, **Nonlinear Projection Operator**.

For each \( \xi \in \text{dom} \mathcal{P} \), the curve \( \eta = \mathcal{P}(\xi) \) is a trajectory.

Note: the trajectory contains both state and control curves.
Introduction

❖ Why do Trajectory Optimization?
❖ Minimization of Trajectory Functionals
❖ Unconstrained (?) Optimal Control
❖ Projection Operator Approach

Projection Operator

❖ Projection Operator Properties
❖ Trajectory Manifold
❖ Equivalent Optimization Problems
❖ Projection operator Newton method
❖ Derivatives
❖ Computation of $D^2 P$

Mathematical Preliminaries

Control systems on Lie groups
The Projection Operator approach on Lie groups
Left-trivialized linearization around a trajectory

Projection Operator

Quadratic approximation of the cost function

$\eta = P(\xi)$
Suppose that $f$ is $C^r$ and that $K$ is bounded and exponentially stabilizes $\xi_0 \in \mathcal{T}$. Then \cite{Hauser1998}

- $\mathcal{P}$ is well defined on an $L_\infty$ neighborhood of $\xi_0$
- $\mathcal{P}$ is $C^r$ (Fréchet diff wrt $L_\infty$ norm)
- $\xi \in \mathcal{T}$ if and only if $\xi = \mathcal{P}(\xi)$
- $\mathcal{P} = \mathcal{P} \circ \mathcal{P}$ (projection)

On the finite interval $[0, T]$, choose $K(\cdot)$ to obtain stability-like properties so that the modulus of continuity of $\mathcal{P}$ is relatively small.

On the infinite horizon, instabilities must be stabilized in order to obtain a projection operator; consider $\dot{x} = x + u$.

**Theorem:** \( \mathcal{T} \) is a *Banach manifold*: Every \( \eta \in \mathcal{T} \) near \( \xi \in \mathcal{T} \) can be uniquely represented as

\[
\eta = P(\xi + \zeta), \quad \zeta \in T_\xi \mathcal{T}
\]

Key: the projection operator \( DP(\xi) \) provides the required *subspace splitting*. Note: \( \zeta \in T_\xi \mathcal{T} \) if and only if \( \zeta = DP(\xi) \cdot \zeta \)
**Equivalent Optimization Problems**

Using the **projection operator**, we see that

$$\min_{\xi \in \mathcal{T}} h(\xi) = \min_{\xi = \mathcal{P}(\xi)} h(\xi)$$

where

$$h(x(\cdot), u(\cdot)) = \int_{0}^{T} l(\tau, x(\tau), u(\tau)) \, d\tau + m(x(T))$$

Furthermore, defining

$$\tilde{h}(\xi) := h(\mathcal{P}(\xi))$$

for $\xi \in \mathcal{U}$ with $\mathcal{P}(\mathcal{U}) \subset \mathcal{U} \subset \text{dom} \mathcal{P}$, we see that

$$\min_{\xi \in \mathcal{T}} h(\xi) \quad \text{and} \quad \min_{\xi \in \mathcal{U}} \tilde{h}(\xi)$$

are **equivalent** in the sense that

- if $\xi^* \in \mathcal{T} \cap \mathcal{U}$ is a **constrained** local minimum of $h$, then it is an **unconstrained** local minimum of $\tilde{h}$;
- if $\xi^+ \in \mathcal{U}$ is an **unconstrained** local minimum of $\tilde{h}$ in $\mathcal{U}$, then $\xi^* = \mathcal{P}(\xi^+)$ is a **constrained** local minimum of $h$. 
**Projection operator Newton method**

**given** initial trajectory $\xi_0 \in \mathcal{T}$

**for** $i = 0, 1, 2, \ldots$

redesign feedback $K(\cdot)$ if desired/needed

descent direction

$$
\zeta_i = \arg \min_{\zeta \in T_{\xi_i} \mathcal{T}} Dh(\xi_i) \cdot \zeta + \frac{1}{2} D^2\tilde{h}(\xi_i) \cdot (\zeta, \zeta) \quad (\text{LQ})
$$

line search

$$
\gamma_i = \arg \min_{\gamma \in (0,1]} h(\mathcal{P}(\xi_i + \gamma \zeta_i))
$$

update

$$
\xi_{i+1} = \mathcal{P}(\xi_i + \gamma_i \zeta_i)
$$

end
**Projection operator Newton method**

Given initial trajectory $\xi_0 \in \mathcal{T}$

for $i = 0, 1, 2, \ldots$

redesign feedback $K(\cdot)$ if desired/needed

descent direction $\zeta_i = \arg\min_{\zeta \in T_{\xi_i} \mathcal{T}} Dh(\xi_i) \cdot \zeta + \frac{1}{2} D^2 \tilde{h}(\xi_i) \cdot (\zeta, \zeta) \quad \text{(LQ)}$

line search $\gamma_i = \arg\min_{\gamma \in (0, 1]} h(\mathcal{P}(\xi_i + \gamma \zeta_i))$

update $\xi_{i+1} = \mathcal{P}(\xi_i + \gamma_i \zeta_i)$

end

This **direct method** generates a descending trajectory sequence in **Banach space**! Also, **quadratic** convergence rate.

Note that

$$h(\xi) + \varepsilon \, Dh(\xi) \cdot \zeta + \frac{1}{2} \, \varepsilon^2 \, D^2 \tilde{h}(\xi) \cdot (\zeta, \zeta)$$

is the **second order approximation** of $\tilde{h}(\xi + \varepsilon \zeta) = h(\mathcal{P}(\xi + \varepsilon \zeta))$

when $\xi \in \mathcal{T}$ and $\zeta \in T_{\xi} \mathcal{T}$. 

---

**Introduction**

- Why do Trajectory Optimization?
- Minimization of Trajectory Functionals
- Unconstrained (?) Optimal Control
- Projection Operator Approach
- Projection Operator
- Projection Operator Properties
- Trajectory Manifold
- Equivalent Optimization Problems
- **Projection operator Newton method**

**Mathematical Preliminaries**

Control systems on Lie groups

The Projection Operator approach on Lie groups

Left-trivialized linearization around a trajectory

Projection Operator

**Derivatives**

**Computation of $D^2 \mathcal{P}$**

Quadratic approximation of the cost function
First and second derivative of $\tilde{h}(\xi) = h(\mathcal{P}(\xi))$ are given by

$$D\tilde{h}(\xi) \cdot \zeta = Dh(\mathcal{P}(\xi)) \cdot D\mathcal{P}(\xi) \cdot \zeta$$

$$D^2\tilde{h}(\xi) \cdot (\zeta_1, \zeta_2) =$$

$$D^2 h(\mathcal{P}(\xi)) \cdot (D\mathcal{P}(\xi) \cdot \zeta_1, D\mathcal{P}(\xi) \cdot \zeta_2)$$

$$+ Dh(\mathcal{P}(\xi)) \cdot D^2\mathcal{P}(\xi) \cdot (\zeta_1, \zeta_2)$$
Derivatives

First and second derivative of \( \tilde{h}(\xi) = h(\mathcal{P}(\xi)) \) are given by

\[
D\tilde{h}(\xi) \cdot \zeta = Dh(\mathcal{P}(\xi)) \cdot D\mathcal{P}(\xi) \cdot \zeta
\]

\[
D^2\tilde{h}(\xi) \cdot (\zeta_1, \zeta_2) = \\
D^2h(\mathcal{P}(\xi)) \cdot (D\mathcal{P}(\xi) \cdot \zeta_1, D\mathcal{P}(\xi) \cdot \zeta_2) \\
+ Dh(\mathcal{P}(\xi)) \cdot D^2\mathcal{P}(\xi) \cdot (\zeta_1, \zeta_2)
\]

When \( \xi \in \mathcal{T} \) and \( \zeta_i \in T_\xi \mathcal{T} \), they specialize into

\[
D\tilde{h}(\xi) \cdot \zeta = Dh(\xi) \cdot \zeta
\]

\[
D^2\tilde{h}(\xi) \cdot (\zeta_1, \zeta_2) = D^2h(\xi) \cdot (\zeta_1, \zeta_2) + Dh(\xi) \cdot D^2\mathcal{P}(\xi) \cdot (\zeta_1, \zeta_2)
\]
First and second derivative of $\tilde{h}(\xi) = h(\mathcal{P}(\xi))$ are given by

$$D\tilde{h}(\xi) \cdot \zeta = Dh(\mathcal{P}(\xi)) \cdot D\mathcal{P}(\xi) \cdot \zeta$$

$$D^2\tilde{h}(\xi) \cdot (\zeta_1, \zeta_2) =$$

$$D^2h(\mathcal{P}(\xi)) \cdot (D\mathcal{P}(\xi) \cdot \zeta_1, D\mathcal{P}(\xi) \cdot \zeta_2)$$

$$+ Dh(\mathcal{P}(\xi)) \cdot D^2\mathcal{P}(\xi) \cdot (\zeta_1, \zeta_2)$$

When $\xi \in \mathcal{T}$ and $\zeta_i \in T_\xi \mathcal{T}$, they specialize into

$$D\tilde{h}(\xi) \cdot \zeta = Dh(\xi) \cdot \zeta$$

$$D^2\tilde{h}(\xi) \cdot (\zeta_1, \zeta_2) = D^2h(\xi) \cdot (\zeta_1, \zeta_2) + Dh(\xi) \cdot D^2\mathcal{P}(\xi) \cdot (\zeta_1, \zeta_2)$$

How to compute $D^2\mathcal{P}(\xi) \cdot (\zeta_1, \zeta_2)$?
Computation of $D^2\mathcal{P}$

We may use ODEs to calculate $D^2\mathcal{P}(\xi) \cdot (\zeta_1, \zeta_2)$:

$$
\begin{align*}
\eta &= (x, u) = \mathcal{P}(\xi) = \mathcal{P}(\alpha, \mu) \\
\gamma_i &= (z_i, v_i) = D\mathcal{P}(\xi) \cdot \zeta_i = D\mathcal{P}(\xi) \cdot (\beta_i, \nu_i) \\
\omega &= (y, w) = D^2\mathcal{P}(\xi) \cdot (\zeta_1, \zeta_2)
\end{align*}
$$

$$
\begin{align*}
\eta(t) : \quad &\dot{x}(t) = f(x(t), u(t)), \quad x(0) = x_0 \\
&u(t) = \mu(t) + K(t)(\alpha(t) - x(t))
\end{align*}
$$

$$
\begin{align*}
\gamma_i(t) : \quad &\dot{z}_i(t) = A(\eta(t))z_i(t) + B(\eta(t))v_i(t), \quad z_i(0) = 0 \\
&v_i(t) = \nu_i(t) + K(t)(\beta_i(t) - z_i(t))
\end{align*}
$$

$$
\begin{align*}
\omega(t) : \quad &\dot{y}(t) = A(\eta(t))y(t) + B(\eta(t))w(t) + D^2f(\eta(t)) \cdot (\gamma_1(t), \gamma_2(t)) \\
&w(t) = -K(t)y(t), \quad y(0) = 0
\end{align*}
$$

- The derivatives are about the trajectory $\eta = \mathcal{P}(\xi)$
- The feedback $K(\cdot)$ stabilizes the state at each level
This was the introduction...

What if the system evolves on a Lie group?
Mathematical Preliminaries

- Smooth manifolds
- Vector fields on a manifold
- Lie groups
- Lie groups (cont'd)
- Lie Algebras
- Triviality and exponential map

Control systems on Lie groups

The Projection Operator approach on Lie groups

Left-trivialized linearization around a trajectory

Projection Operator

Quadratic approximation of the cost function
A smooth manifold $M$ is a set which "locally looks like $\mathbb{R}^n$". Think about, e.g., the 2-sphere $S^2$ in $\mathbb{R}^3$.

- Manifolds with be indicated with capital letters, usually $M$ or $N$.
- A point on the manifold will be denoted simply by $x$.
- $T_xM$ and $T_x^*M$ denote, respectively, the tangent and cotangent spaces of $M$ at $x$.
- A generic tangent vector is usually written as $v_x$ or $w_x$.
- The tangent and cotangent bundles of $M$ are denoted by $TM$ and $T^*M$, respectively.
Vector fields on a manifold

- The natural bundle projection from $TM$ to $M$ is the mapping
  \[ \pi : \quad TM \quad \to \quad M \]
  \[ v_x \quad \mapsto \quad x \]

- A vector field on a manifold $M$ is a mapping
  \[ X : \quad M \quad \to \quad TM \]
  \[ x \quad \mapsto \quad X(x) \]
  which is a section of the tangent bundle $TM$, that is, it satisfies
  \[ \pi X(x) = x \]
Lie groups

- A **Lie group** is a smooth manifold endowed with a group structure. The group operation must be **smooth**.

- A generic **Lie group** is denoted by $G$.

- Typical examples are the groups $SO(3)$, $SE(2)$, $SE(3)$, and $U(n)$...

- ...but also $T SO(3)$, $T SE(2)$, $T SE(3)$ are Lie groups!

  These are called the **tangent groups**. Our theory apply to mechanical systems.
Lie groups (cont’d)

- **Left and right translations** of \( x \in G \) (a group element) by the group element \( g \in G \) are denoted by
  
  \[ L_gx \quad \text{and} \quad R_gx, \]

  respectively.

- When convenient, we will adopt also the **shorthand notation**
  
  \[ gx, \quad xg, \quad gv_x, \quad v_xg \]

  for, in the same order,

  \[ L_gx, \quad R_gx, \quad T_xL_g(v_x), \quad T_xR_g(v_x) \]

  .
A left-invariant vector field on $G$ is a vector field $X$ that satisfies

$$X(L_g x) = (T_x L_g) X(x).$$

Given $\varrho \in T_e G$, the symbol $X_\varrho$ is the associated left-invariant vector field

$$X_\varrho(g) := T_e L_g(\varrho).$$

The Lie algebra $\mathfrak{g}$ is identified with the tangent space $T_e G$ endowed with the Lie bracket operation

$$[\cdot, \cdot] : \mathfrak{g} \times \mathfrak{g} \to \mathfrak{g},$$

defined by

$$[\varrho, \varsigma] := [X_\varrho, X_\varsigma](e),$$

where the later bracket is the Jacobi-Lie bracket evaluated at the group identity.
The tangent bundle $TG$ of Lie groups $G$ is trivial. That is,

$$TG \approx G \times \mathfrak{g}.$$ 

The exponential map $\exp : \mathfrak{g} \rightarrow G$ is a diffeomorphism between a neighborhood of the origin of the Lie Algebra $\mathfrak{g}$ and a neighborhood of the identity of the Lie group $G$.

The exponential map $\exp : \mathfrak{g} \rightarrow G$ can be used to parameterize the neighborhood of any point $g \in G$.

Using left translation, we parameterize a neighborhood of $g \in G$ as

$$g \exp(\xi), \quad \xi \in \mathfrak{g}$$
The tangent bundle $TG$ of Lie groups $G$ is trivial. That is,

$$TG \approx G \times \mathfrak{g}.$$ 

The **exponential map** $\exp : \mathfrak{g} \to G$ is a diffeomorphism between a neighborhood of the origin of the Lie Algebra $\mathfrak{g}$ and a neighborhood of the identity of the Lie group $G$.

The exponential map $\exp : \mathfrak{g} \to G$ can be used to parameterize the neighborhood of any point $g \in G$.

Using left translation, we parameterize a neighborhood of $g \in G$ as

$$g \exp(\xi), \quad \xi \in \mathfrak{g}$$

**Key idea:** On a Lie group, the expansion of a function $f : G_1 \to G_2$ is written as

$$f(g \exp_{G_1}(tv)) = f(g) \exp_{G_2}(n_v(t)).$$

This generalized on a vector space

$$f(x + tv) = f(g) + n_v(t)$$
Control systems on Lie groups

- Control systems on a Lie group
- The Projection Operator approach on Lie groups
- Left-trivialized linearization around a trajectory
- Projection Operator
- Quadratic approximation of the cost function
A control system on a Lie group $G$ is a mapping

$$f : G \times \mathbb{R}^m \times \mathbb{R} \rightarrow TG$$

$$(g, u, t) \mapsto f(g, u, t),$$

such that $\pi f(g, u, t) = g$ for each $(g, u, t) \in G \times \mathbb{R}^m \times \mathbb{R}$.

A state trajectory $g(t), t \geq 0$, of $f$ is an absolutely continuous curve in $G$ that satisfies (a.e.), for an assigned input $u(t)$,

$$\dot{g}(t) = f(g(t), u(t), t).$$

We will assume $f$ is sufficiently smooth, Lipschitz, ... to guarantee existence and uniqueness of solutions.

We can rewrite $\dot{g}(t) = f(g(t), u(t), t)$ as

$$\dot{g}(t) = g(t)\lambda(g(t), u(t), t),$$

where $\lambda : G \times \mathbb{R}^m \times \mathbb{R} \rightarrow g$, $\lambda(g, u, t) := g^{-1}f(g, u, t)$ is the left trivialization of the control vector field $f$. 

The Projection Operator approach on Lie groups

❖ Minimization of Trajectory Functionals
❖ Projection operator
Newton method

Left-trivialized linearization around a trajectory

Projection Operator
Quadratic approximation of the cost function
Consider the problem of minimizing a functional

\[ h(g(\cdot), u(\cdot)) := \int_0^T l(g(\tau), u(\tau), \tau) \, d\tau + m(g(T)) \]

over the set \( \mathcal{T} \) of (bounded) trajectories of the nonlinear system

\[ \dot{g}(t) = f(x(t), u(t)) = g\lambda(g(t), u(t)) \]

with \( g(0) = g_0 \).

As in the vector case, we write this constrained problem as

\[ \min_{\xi \in \mathcal{T}} h(\xi) \]

where \( \xi = (\alpha(\cdot), \mu(\cdot)) \) is in general a (bounded) curve with \( \alpha(\cdot) \) continuous and \( \alpha(0) = g_0 \).
Consider the problem of minimizing a functional

\[ h(g(\cdot), u(\cdot)) := \int_0^T l(g(\tau), u(\tau), \tau) \, d\tau + m(g(T)) \]

over the set \( \mathcal{T} \) of (bounded) trajectories of the nonlinear system

\[ \dot{g}(t) = f(x(t), u(t)) = g\lambda(g(t), u(t)) \]

with \( g(0) = g_0 \).

As in the vector case, we write this **constrained** problem as

\[ \min_{\xi \in \mathcal{T}} h(\xi) \]

where \( \xi = (\alpha(\cdot), \mu(\cdot)) \) is in general a (bounded) curve with \( \alpha(\cdot) \) continuous and \( \alpha(0) = g_0 \).

**How can we generalize the Projection Operator approach to Lie groups?**
The Newton algorithm is structurally the same:

**given** initial trajectory $\xi_0 \in \mathcal{T}$

**for** $i = 0, 1, 2, \ldots$

redesign feedback $K(\cdot)$ if desired/needed

descent direction

$$\zeta_i = \arg \min_{\xi_i \zeta \in T_{\xi_i} \mathcal{T}} Dh(\xi_i) \cdot \xi_i \zeta + \frac{1}{2} D^2 \tilde{h}(\xi_i) \cdot (\xi_i \zeta, \xi_i \zeta) \quad (LQ)$$

line search

$$\gamma_i = \arg \min_{\gamma \in (0,1]} h(P(\xi_i \exp(\gamma \zeta_i)))$$

update

$$\xi_{i+1} = P(\xi_i \exp(\gamma_i \zeta))$$

**end**
**Projection operator Newton method**

The Newton algorithm is structurally the same:

**given** initial trajectory $\xi_0 \in \mathcal{T}$

**for** $i = 0, 1, 2, \ldots$

redesign feedback $K(\cdot)$ if desired/needed

**descent direction**

$$\zeta_i = \arg \min_{\xi_i \zeta \in T_{\xi_i} \mathcal{T}} Dh(\xi_i) \cdot \xi_i \zeta + \frac{1}{2} D^2 \tilde{h}(\xi_i) \cdot (\xi_i \zeta, \xi_i \zeta) \quad \text{(LQ)}$$

**line search**

$$\gamma_i = \arg \min_{\gamma \in (0,1]} h(\mathcal{P}(\xi_i \exp(\gamma \zeta_i)))$$

**update**

$$\xi_{i+1} = \mathcal{P}(\xi_i \exp(\gamma_i \zeta_i))$$

**end**

- What is the linearization of a system evolving of a Lie group? $\xi_i \zeta \in T_{\xi_i} \mathcal{T}$.

- What does it mean to compute a second derivative on a Lie groups? $Dh(\xi_i) \cdot \xi_i \zeta + \frac{1}{2} D^2 \tilde{h}(\xi_i) \cdot (\xi_i \zeta, \xi_i \zeta)$.
Left-trivialized linearization around a trajectory
Let \( \eta(t) = (g(t), u(t)), \quad t \in [0, \infty) \) be a the state-input trajectory of \( f \).

Consider the **linear perturbation** of the input defined as
\[
u_\varepsilon(t) := u(t) + \varepsilon v(t)\]

Indicate with \( g_\varepsilon \) the **perturbed state trajectory** associated with \( u_\varepsilon \).

The state trajectory \( g_\varepsilon \) satisfies
\[
\dot{g}_\varepsilon(t) = g_\varepsilon(t) \lambda(g_\varepsilon(t), u_\varepsilon(t), t), \\
g_\varepsilon(0) = g_0.
\]
Define the **left-trivialized perturbed trajectory**

$$z_\varepsilon(t), \quad t \in [0, T(\varepsilon)),$$

so that

$$g_\varepsilon(t) = g(t) \exp(z_\varepsilon(t)), \quad t \in [0, T(\varepsilon))$$

Define $$x_\varepsilon(t) := \exp z_\varepsilon(t).$$

The left trivialized perturbed trajectory satisfies

$$\dot{z}_\varepsilon = d \log_{z_\varepsilon} \left( \text{Ad}_{x_\varepsilon} \lambda(gx_\varepsilon, u_\varepsilon, t) - \lambda(g, u, t) \right)$$

$$z_\varepsilon(0) = 0.$$

where

$$d \log_{g} \varsigma = D \log(\exp(g)) \cdot \exp(g) \varsigma \quad \text{(trivialized tangent)}$$

and

$$\text{Ad}$$ is the **adjoint action** of $$G$$ on $$g.$$
The left-trivialized perturbed trajectory $z_\varepsilon(t), t \geq 0$, can be expanded to first order as $z_\varepsilon(t) = \varepsilon z(t) + o(\varepsilon)$, where $z(t)$ is given by the left-trivialized linearization

$$
\dot{z}(t) = A(\eta(t), t) z(t) + B(\eta(t), t) v(t), \\
z(0) = z_0,
$$

with

$$
A(\eta, t) := D_1 \lambda(g, u, t) \circ TL_g - \text{ad}_{\lambda(g, u, t)}, \\
B(\eta, t) := D_2 \lambda(g, u, t),
$$

where $\text{ad}$ is the adjoint action of $g$ on itself.
Projection Operator

Projection Operator on a Lie Group

Linearization of the Projection Operator

Quadratic approximation of the cost function
Projection Operator on a Lie Group

- **Vector space** $\mathbb{R}^n$
  The Projection Operator $\eta = (x, u) = \mathcal{P}(\alpha, \mu) = \mathcal{P}(\xi)$ is given by
  \[
  \dot{x} = f(x, k(x, \xi, t)) \\
  u = k(x, \xi, t) = \alpha + K(t)(\mu - x)
  \]

- **Lie group** $G$
  The Projection Operator $\eta = (x, u) = \mathcal{P}(\alpha, \mu) = \mathcal{P}(\xi)$ is given by
  \[
  \dot{g} = f(g, k(g, \xi, t)) = g\lambda(g, k(g, \xi, t)) \\
  u = k(g, \xi, t) = \alpha + K(t)[\log(g^{-1}\mu)]
  \]
**Projection Operator on a Lie Group**

- **Vector space** $\mathbb{R}^n$
  The Projection Operator $\eta = (x, u) = \mathcal{P}(\alpha, \mu) = \mathcal{P}(\xi)$ is given by
  
  $$\dot{x} = f(x, k(x, \xi, t))$$
  $$u = k(x, \xi, t) = \alpha + K(t)(\mu - x)$$

- **Lie group** $G$
  The Projection Operator $\eta = (x, u) = \mathcal{P}(\alpha, \mu) = \mathcal{P}(\xi)$ is given by
  
  $$\dot{g} = f(g, k(g, \xi, t)) = g\lambda(g, k(g, \xi, t))$$
  $$u = k(g, \xi, t) = \alpha + K(t)[\log(g^{-1}\mu)]$$

- Note that $(\mathbb{R}^n, +)$ is an abelian Lie group!
  Given $x_1, x_2 \in \mathbb{R}^n$, $x_2^{-1}x_1 = x_1 - x_2 = -x_2 + x_1$.
  Also, $\exp(v) = v$, $\text{Ad} = \text{id}$, and $\text{ad} = \text{id}$.

  The theory on $\mathbb{R}^n$ is a **special case** of the general theory!
Linearization of the Projection Operator

Introduction

Mathematical Preliminaries

Control systems on Lie groups

The Projection Operator approach on Lie groups

Left-trivialized linearization around a trajectory

Projection Operator

❖ Projection Operator on a Lie Group

❖ Linearization of the Projection Operator

Quadratic approximation of the cost function

<table>
<thead>
<tr>
<th>Curve</th>
<th>$\mathbf{\xi} = (\alpha, \mu)$</th>
<th>$\mathbb{R}^n \times \mathbb{R}^m$</th>
<th>Vector Space</th>
</tr>
</thead>
<tbody>
<tr>
<td>Perturbation</td>
<td>$\mathbf{\zeta} = (\beta, \nu)$</td>
<td>$\mathbb{R}^n \times \mathbb{R}^m$</td>
<td>Lie Group</td>
</tr>
<tr>
<td>Trajectory</td>
<td>$\mathbf{\eta} = (g, u)$</td>
<td>$\mathbb{R}^n \times \mathbb{R}^m$</td>
<td></td>
</tr>
<tr>
<td>Traj. perturbation</td>
<td>$\mathbf{\gamma} = (z, v)$</td>
<td>$\mathbb{R}^n \times \mathbb{R}^m$</td>
<td></td>
</tr>
</tbody>
</table>

- Vector space $\mathbb{R}^n$

$\mathcal{P}(\mathbf{\xi} + \varepsilon \mathbf{\zeta}) = \mathbf{\eta} + \varepsilon \mathbf{\gamma} + o(\varepsilon)$. We obtain

$$
\begin{aligned}
\dot{z} &= A(\eta(t))z + B(\eta(t))v, \\
v &= \nu + K(t)(\beta - z)
\end{aligned}
$$

- Lie group $\mathbb{G}$

$\mathcal{P}(\mathbf{\xi} \exp(\varepsilon \mathbf{\zeta})) = \mathcal{P}(\mathbf{\xi}) \exp(\varepsilon \mathbf{\gamma} + o(\varepsilon))$. We obtain, recall $\mathcal{P}(\mathbf{\xi}) = \mathbf{\eta}$,

$$
\begin{aligned}
\dot{z} &= A(\eta(t))z + B(\eta(t))v, \\
v &= \nu + K(t)d\log_{\log(g^{-1}\alpha)}(Ad_{g^{-1}\alpha}\beta - z)
\end{aligned}
$$
Linearization of the Projection Operator

**Vector Space** \( \mathbb{R}^n \times \mathbb{R}^m \)

**Lie Group** \( G \times \mathbb{R}^m \)

<table>
<thead>
<tr>
<th>Curve</th>
<th>( \xi = (\alpha, \mu) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Perturbation</td>
<td>( \zeta = (\beta, \nu) )</td>
</tr>
<tr>
<td>Trajectory</td>
<td>( \eta = (g, u) )</td>
</tr>
<tr>
<td>Traj. perturbation</td>
<td>( \gamma = (z, v) )</td>
</tr>
</tbody>
</table>

- **Vector space** \( \mathbb{R}^n \)

  \( \mathcal{P}(\xi + \varepsilon \zeta) = \eta + \varepsilon \gamma + o(\varepsilon) \). We obtain

  \[
  \dot{z} = A(\eta(t))z + B(\eta(t))v, \quad z(0) = 0
  \]

  \[
  v = \nu + K(t)(\beta - z)
  \]

- **Lie group** \( G \)

  \( \mathcal{P}(\xi \exp(\varepsilon \zeta)) = \mathcal{P}(\xi) \exp(\varepsilon \gamma + o(\varepsilon)) \). We obtain, recall \( \mathcal{P}(\xi) = \eta \),

  \[
  \dot{z} = A(\eta(t))z + B(\eta(t))v, \quad z(0) = 0
  \]

  \[
  v = \nu + K(t)\text{dlog}_\log(g^{-1}\alpha)(\text{Ad}_{g^{-1}\alpha}\beta - z)
  \]

When \( \xi = \mathcal{P}(\xi) = \eta \), \( \text{dlog}_\log(g^{-1}\alpha) = \text{id} \) and \( \text{Ad}_{g^{-1}\alpha} = \text{id} \)!
Quadratic approximation of the cost function
We can expand \( \tilde{h}(\xi \exp(\varepsilon \zeta)) := h(\mathcal{P}(\xi \exp(\varepsilon \zeta))) \) as

\[
\tilde{h}(\xi \exp(\varepsilon \zeta)) = h(\mathcal{P}(\xi)) + \varepsilon \, D\tilde{h}(\xi) \cdot \xi \zeta \\
+ \frac{1}{2} \varepsilon^2 \, D^2 \tilde{h}(\xi) \cdot (\xi \zeta, \xi \zeta) + o(\varepsilon^2)
\]
We can expand $\tilde{h}(\xi \exp(\varepsilon \zeta)) := h(\mathcal{P}(\xi \exp(\varepsilon \zeta)))$ as

$$
\tilde{h}(\xi \exp(\varepsilon \zeta)) = h(\mathcal{P}(\xi)) + \varepsilon D\tilde{h}(\xi) \cdot \xi \zeta
+ 1/2 \varepsilon^2 D^2\tilde{h}(\xi) \cdot (\xi \zeta, \xi \zeta) + o(\varepsilon^2)
$$

First and second derivative of $\tilde{h}(\xi) = h(\mathcal{P}(\xi))$ are given by

$$
D\tilde{h}(\xi) \cdot \xi \zeta = Dh(\mathcal{P}(\xi)) \cdot D\mathcal{P}(\xi) \cdot \xi \zeta
$$

$$
D^2\tilde{h}(\xi) \cdot (\xi \zeta_1, \xi \zeta_2) =
D^2h(\mathcal{P}(\xi)) \cdot (D\mathcal{P}(\xi) \cdot \xi \zeta_1, D\mathcal{P}(\xi) \cdot \xi \zeta_2)
+ Dh(\mathcal{P}(\xi)) \cdot D^2\mathcal{P}(\xi) \cdot (\xi \zeta_1, \xi \zeta_2)
$$
We can expand \( \tilde{h}(\xi \exp(\varepsilon \zeta)) := h(\mathcal{P}(\xi \exp(\varepsilon \zeta))) \) as

\[
\tilde{h}(\xi \exp(\varepsilon \zeta)) = h(\mathcal{P}(\xi)) + \varepsilon D\tilde{h}(\xi) \cdot \xi \zeta \\
+ 1/2 \varepsilon^2 D^2\tilde{h}(\xi) \cdot (\xi \zeta, \xi \zeta) + o(\varepsilon^2)
\]

First and second derivative of \( \tilde{h}(\xi) = h(\mathcal{P}(\xi)) \) are given by

\[
D\tilde{h}(\xi) \cdot \xi \zeta = Dh(\mathcal{P}(\xi)) \cdot D\mathcal{P}(\xi) \cdot \xi \zeta
\]

\[
D^2\tilde{h}(\xi) \cdot (\xi \zeta_1, \xi \zeta_2) = \\
D^2h(\mathcal{P}(\xi)) \cdot (D\mathcal{P}(\xi) \cdot \xi \zeta_1, D\mathcal{P}(\xi) \cdot \xi \zeta_2) \\
+ Dh(\mathcal{P}(\xi)) \cdot D^2\mathcal{P}(\xi) \cdot (\xi \zeta_1, \xi \zeta_2)
\]

When \( \xi \in \mathcal{T} \) and \( \xi \zeta_i \in T_{\xi} \mathcal{T} \), they specialize into

\[
D\tilde{h}(\xi) \cdot \xi \zeta = Dh(\xi) \cdot \xi \zeta
\]

\[
D^2\tilde{h}(\xi) \cdot (\xi \zeta_1, \xi \zeta_2) = D^2h(\xi) \cdot (\xi \zeta_1, \xi \zeta_2) + Dh(\xi) \cdot D^2\mathcal{P}(\xi) \cdot (\xi \zeta_1, \xi \zeta_2)
\]
We can expand $\tilde{h}(\xi \exp(\varepsilon \zeta)) := h(\mathcal{P}(\xi \exp(\varepsilon \zeta)))$ as
$$
\tilde{h}(\xi \exp(\varepsilon \zeta)) = h(\mathcal{P}(\xi)) + \varepsilon D\tilde{h}(\xi) \cdot \xi \zeta + \frac{1}{2} \varepsilon^2 \mathbb{D}^2\tilde{h}(\xi) \cdot (\xi \zeta, \xi \zeta) + o(\varepsilon^2)
$$

First and second derivative of $\tilde{h}(\xi) = h(\mathcal{P}(\xi))$ are given by
$$
D\tilde{h}(\xi) \cdot \xi \zeta = Dh(\mathcal{P}(\xi)) \cdot D\mathcal{P}(\xi) \cdot \xi \zeta
$$
$$
\mathbb{D}^2\tilde{h}(\xi) \cdot (\xi \zeta_1, \xi \zeta_2) = 
\mathbb{D}^2 h(\mathcal{P}(\xi)) \cdot (D\mathcal{P}(\xi) \cdot \xi \zeta_1, D\mathcal{P}(\xi) \cdot \xi \zeta_2) + Dh(\mathcal{P}(\xi)) \cdot \mathbb{D}^2 \mathcal{P}(\xi) \cdot (\xi \zeta_1, \xi \zeta_2)
$$

When $\xi \in \mathcal{T}$ and $\xi \zeta_i \in T_{\xi} \mathcal{T}$, they specialize into
$$
D\tilde{h}(\xi) \cdot \xi \zeta = Dh(\xi) \cdot \xi \zeta
$$
$$
\mathbb{D}^2\tilde{h}(\xi) \cdot (\xi \zeta_1, \xi \zeta_2) = \mathbb{D}^2 h(\xi) \cdot (\xi \zeta_1, \xi \zeta_2) + Dh(\xi) \cdot \mathbb{D}^2 \mathcal{P}(\xi) \cdot (\xi \zeta_1, \xi \zeta_2)
$$

How to compute $\mathbb{D}^2 \mathcal{P}(\xi) \cdot (\xi_1, \xi_2)$?
Second order approximation of the Projection Operator

- **Vector space** $\mathbb{R}^n$.

  \[ \omega = D\mathcal{P}^2(\xi) \cdot (\zeta_1, \zeta_2) \]

  with $\xi \in \mathcal{T}$ and $\gamma_i = D\mathcal{P}(\xi) \cdot \zeta_i$,

  \[
  \dot{y} = A(\eta) y + B(\eta) w + D^2 \lambda(\eta) \cdot (\gamma_1, \gamma_2), \quad y(0) = 0, \\
  w = -K(t) y,
  \]

- **Lie group** $G$.

  \[ \mathcal{P}(\xi) \omega = D\mathcal{P}^2(\xi) \cdot (\xi \zeta_1, \xi \zeta_2) \]

  with $\xi \in \mathcal{T}$ and $\mathcal{P}(\xi) \gamma_i = D\mathcal{P}(\xi) \cdot \xi \zeta_i$,

  \[
  \dot{y} = A(\eta) y + B(\eta) w \\
  - 1/2 \left[(\text{ad}_{\zeta_1} \text{ad}_{\zeta_2} + \text{ad}_{\zeta_2} \text{ad}_{\zeta_1}) \lambda(\eta) \\
  - \text{ad}_{\zeta_1} (A(\eta) z_2 + B(\eta) v_2) \\
  - \text{ad}_{\zeta_2} (A(\eta) z_1 + B(\eta) v_1) \right] \\
  + D^2 \lambda(\eta) \cdot (\eta \gamma_1, \eta \gamma_2), \quad y(0) = 0, \\
  w = -K(t) \left[y + 1/2 \left([z_1, \beta_2] + [z_2, \beta_1]\right)\right]
  \]

Recall $\gamma_i = (z_i, v_i)$, $\zeta_i = (\beta_i, \nu_i)$. 

\[ \text{Derivatives} \]
\[ \text{Second geometric derivative} \]
\[ \text{Second geometric derivative (cont'd)} \]
\[ \text{Conclusions} \]
Second geometric derivative

Let $M_1$ and $M_2$ be two smooth manifolds endowed with affine connections $\nabla^1$ and $\nabla^2$, respectively. Let $f : M_1 \to M_2$ be a smooth mapping.

The second geometric derivative is a tool to extend the classical (Leibniz’s) product rule to the covariant derivative of the “product” $Df(\gamma_1(t)) \cdot V_1(t)$, for a curve $\gamma_1$ and a vector field $V_1$ along $\gamma_1$ in $M_1$.

Chosen $x \in M_1$ and two tangent vectors $v_x$ and $w_x \in T_x M_1$. Let $\gamma_1 : I \to M_1$ be a smooth curve in $M_1$ such that

$$\gamma_1(t_0) = x \quad \text{and} \quad \dot{\gamma}_1(t_0) = w_x.$$ 

Let $V_1$ a smooth vector field along $\gamma_1$ such that

$$V_1(t_0) = v_x,$$

and

$$V_2(t) := Df(\gamma_1(t)) \cdot V_1(t) \in T_{f(\gamma_1(t))} M_2,$$

a smooth vector field along the curve $\gamma_2(t) := f(\gamma_1(t))$ in $M_2$. 
**Second geometric derivative (cont’d)**

The **second geometric derivative** of the map \( f : M_1 \rightarrow M_2 \) at \( x \in M_1 \) in the directions \( v_x \) and \( w_x \in T_x M_1 \) is the bilinear mapping \( \mathbb{D}^2 f(x) : T_x M_1 \times T_x M_1 \rightarrow T_{f(x)} M_2 \) defined as

\[
\mathbb{D}^2 f(x) \cdot (v_x, w_x) := D_t V_2(t_0) - D f(\gamma_1(t_0)) \cdot D_t V_1(t_0),
\]

where \( D_t V_1 \) and \( D_t V_2 \) denote the covariant differentiation with respect to \( ^1\nabla \) and \( ^2\nabla \), respectively.
The second geometric derivative of the map $f : M_1 \rightarrow M_2$ at $x \in M_1$ in the directions $v_x$ and $w_x \in T_xM_1$ is the bilinear mapping $\mathcal{D}^2 f(x) : T_xM_1 \times T_xM_1 \rightarrow T_{f(x)}M_2$ defined as

$$\mathcal{D}^2 f(x) \cdot (v_x, w_x) := D_t V_2(t_0) - Df(\gamma_1(t_0)) \cdot D_t V_1(t_0),$$

(1)

where $D_t V_1$ and $D_t V_2$ denote the covariant differentiation with respect to $\nabla^1$ and $\nabla^2$, respectively.

Denote by $P^1$ and $P^2$ the parallel displacements associated to $\nabla^1$ and $\nabla^2$, respectively. Then, equation (1) is equal (for $t = t_0$) to

$$\lim_{\varepsilon \to 0} \frac{1}{\varepsilon} \left( P^2_{\gamma_2 \gamma_1^{-t+\varepsilon}} Df(\gamma_1(t + \varepsilon)) \cdot P^1_{\gamma_1^{-t+\varepsilon}} X_1(\gamma_1(t)) - Df(\gamma_1(t)) \cdot X_1(\gamma_1(t)) \right),$$

(2)
The **second geometric derivative** of the map $f : M_1 \to M_2$ at $x \in M_1$ in the directions $v_x$ and $w_x \in T_x M_1$ is the bilinear mapping

$$D^2 f(x) : T_x M_1 \times T_x M_1 \to T_{f(x)} M_2$$

defined as

$$D^2 f(x) \cdot (v_x, w_x) := D_{t_0} V_2(t_0) - D f(\gamma_1(t_0)) \cdot D_{t_0} V_1(t_0),$$

where $D_{t_0} V_1$ and $D_{t_0} V_2$ denote the covariant differentiation with respect to $^1\nabla$ and $^2\nabla$, respectively.

Denote by $^1P$ and $^2P$ the parallel displacements associated to $^1\nabla$ and $^2\nabla$, respectively. Then, equation (1) is equal (for $t = t_0$) to

$$\lim_{\varepsilon \to 0} \frac{1}{\varepsilon} \left( ^2P_{\gamma_2}^{\gamma_1(t + \varepsilon)} D f(\gamma_1(t + \varepsilon)) \cdot ^1P_{\gamma_1}^{t + \varepsilon} X_1(\gamma_1(t)) - D f(\gamma_1(t)) \cdot X_1(\gamma_1(t)) \right),$$

Those concepts need to be specialized for Lie groups. We used the **symmetric (0)-Cartan-Shouten connection**... no time for the details, unfortunately!
Conclusions

we have extended the **projection operator based trajectory optimization approach** to the class of nonlinear systems that evolve on **non-compact Lie groups** [2].

This required the introduction of a **geometric derivative** notion for the repeated differentiation of a mapping between two Lie groups, endowed with affine connections. *(Not explained for time constraints...)*

With this tool, chain rule like formulas were used to develop expressions for the basic objects needed for trajectory optimization.

**Coding of the algorithm and numerical tests are under development!**

Obrigado pela vossa atenção!